Incorporating Commonsense Knowledge into Story Ending Generation via Heterogeneous Graph Networks

Jiaan Wang[1], Beiqi Zou[3], Zhixu Li[2], Jianfeng Qu[1](✉), Pengpeng Zhao[1], An Liu[1], and Lei Zhao[1]

[1] School of Computer Science and Technology, Soochow University, Suzhou, China
jawang1@stu.suda.edu.cn, {jfqu,ppzhao,anliu,zhaol}@suda.edu.cn
[2] Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai, China
zhixuli@fudan.edu.cn
[3] Department of Computer Science, Princeton University, Princeton, USA
bzou@cs.princeton.edu

Abstract. Story ending generation is an interesting and challenging task, which aims to generate a coherent and reasonable ending given a story context. The key challenges of the task lie in how to comprehend the story context sufficiently and handle the implicit knowledge behind story clues effectively, which are still under-explored by previous work. In this paper, we propose a Story Heterogeneous Graph Network (SHGN) to explicitly model both the information of story context at different granularity levels and the multi-grained interactive relations among them. In detail, we consider commonsense knowledge, words and sentences as three types of nodes. To aggregate non-local information, a global node is also introduced. Given this heterogeneous graph network, the node representations are updated through graph propagation, which adequately utilizes commonsense knowledge to facilitate story comprehension. Moreover, we design two auxiliary tasks to implicitly capture the sentiment trend and key events lie in the context. The auxiliary tasks are jointly optimized with the primary story ending generation task in a multi-task learning strategy. Extensive experiments on the ROCStories Corpus show that the developed model achieves new state-of-the-art performances. Human study further demonstrates that our model generates more reasonable story endings.

Keywords: Story ending generation · Heterogeneous graph network · Multi-task learning

1 Introduction

Story ending generation (SEG) is a natural language generation task, which aims at concluding a story ending given a context [33]. Generally, a story con-

J. Wang and B. Zou—Indicates equal contribution.

A. Bhattacharya et al. (Eds.): DASFAA 2022, LNCS 13247, pp. 85–100, 2022.
https://doi.org/10.1007/978-3-031-00129-1_6

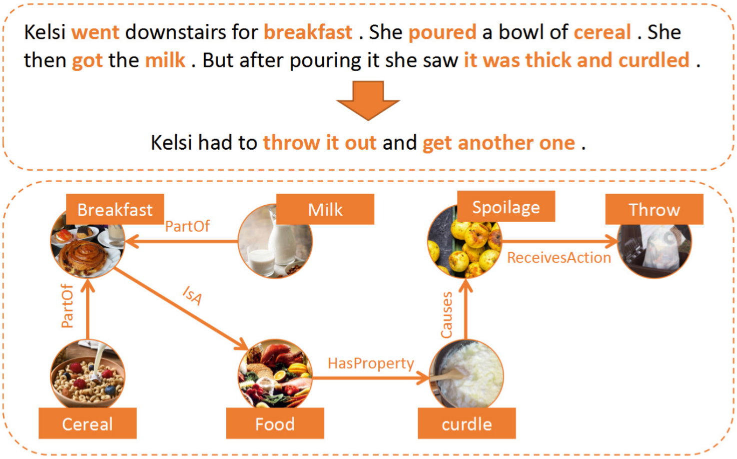

Kelsi went downstairs for breakfast . She poured a bowl of cereal . She then got the milk . But after pouring it she saw it was thick and curdled .
Kelsi had to throw it out and get another one .
Breakfast
PartOf
Milk
Spoilage
Throw
ReceivesAction
PartOf
IsA
Causes
HasProperty
Cereal
Food
curdle

Fig. 1. The top graph shows an example of story ending generation. Orange words are entities and events. The bottom graph indicates the implicit knowledge behind the story. (Color figure online)

text contains a series of entities and events (known as story clues), each of which could have strong logical relationships with others, which leads to rich interactive relations across the whole context. For humans, one may utilize his/her own commonsense knowledge to capture the story clues and conceive story endings. As shown in Fig. 1, the story clue of example story is: *went_breakfast* ⇒ *poured_cereal* ⇒ *got_milk* ⇒ *it_was_thick_and_curdled*, which indicates the food may have gone bad. It is natural for humans to *throw the bad food and get another one* since we all know *spoiled food is harmful to health*. Therefore, in order to generate a coherent and reasonable ending, generative models should not only sufficiently comprehend the story context and further capture the story clues but also effectively handle the implicit knowledge behind them.

Most previous work views the story context as a linear sequence of words and ignores the rich relations among them. For example, Zhao et al. [33] and Li et al. [13] explore variant sequence-to-sequence (Seq2Seq) models encoding the story context in a left-to-right manner and decoding endings. They further utilize reinforcement learning to improve the rationality and/or diversity of generated endings. Gupta et al. [7] resort to an extra keywords extraction algorithm and model the keywords information in the Seq2Seq models. Luo et al. [15] make use of existing sentiment analyzers to consider the fine-grained sentiment in their Seq2Seq method. Guan et al. [6] design incremental encoding and multi-source attention mechanism to model the relation of adjacent sentences and incorporate commonsense knowledge into contextual representation. These methods all adopt sequential modeling strategies for encoding the story context, hindering the exploration of inherently rich interactive relations across the whole context, which makes the story context and commonsense knowledge modeling inadequate.

Recently, Huang et al. [10] suggest that the great importance of story clues hidden in the context and further propose multi-level graph convolutional networks over dependency parse (MGCN-DP) that models SEG task in a graph-to-text manner. The graph architecture better models the interactive relations across story context and result in more coherent endings compared to sequential methods. However, the relations of words from different context sentences cannot be explicitly captured by MGCN-DP which only contains word nodes from the same sentence in a graph. Besides, the MGCN-DP model does not consider commonsense knowledge behind the story, thus the generated endings could be suboptimal.

To remedy above issues, in this paper, we propose a **S**tory **H**eterogeneous **G**raph **N**etwork (SHGN) for SEG. The heterogeneous graph network shows its superiority in many tasks, such as recommender systems and summarization [2,29]. Specifically, three types of graph nodes are considered in our SHGN: *commonsense knowledge*, *words* and *sentences*. Besides, a *global* node is also introduced to the graph to aggregate the non-local information (see Fig. 3). To obtain contextualized representations for these nodes, large-scale pre-trained embedding models such as Sentence-BERT [22] and SimCSE [4] are used for contextual encoding. Then, a graph neural network is used to propagate message and update representations of nodes in the heterogeneous graph. The final node representations are passed through transformer decoders to generate story endings. In addition, to sufficiently comprehend the story context, we design two sub-tasks with special consideration for story ending generation: (1) the *sentiment prediction of story endings sub-task* uses representations of sentence nodes to predict the sentiment of corresponding story ending, which is constructed to push the model to capture the fine-grained sentiment trend; (2) the *clue words prediction sub-task* utilizes representations of word nodes to predict whether each word belongs to story clues, which is expected to force the model to identify the key events in the context. Such two sub-tasks can be coupled with the primary story ending generation task via multi-task learning strategy, resulting in our final model SHGN.

We conduct various experiments on the widely used ROCStories Corpus [19]. Experimental results show that our approach achieves state-of-the-art performances on SEG task. Human study indicates that our SHGN generates more coherent and reasonable story endings as compared to previous strong baselines.

Our main contributions in this paper are summarized as follows:

- We propose a Story Heterogeneous Graph Network for SEG, which explicitly models the information of story context at different granularity levels and the multi-grained interactive relations among them[1].
- We also design two auxiliary tasks (i.e., sentiment prediction of story endings and clue words prediction) to facilitate story comprehension. To the best of our knowledge, we are the first to apply multi-task learning strategy on SEG.

[1] We release our code and generated results at https://github.com/krystalan/AwesomeSEG.

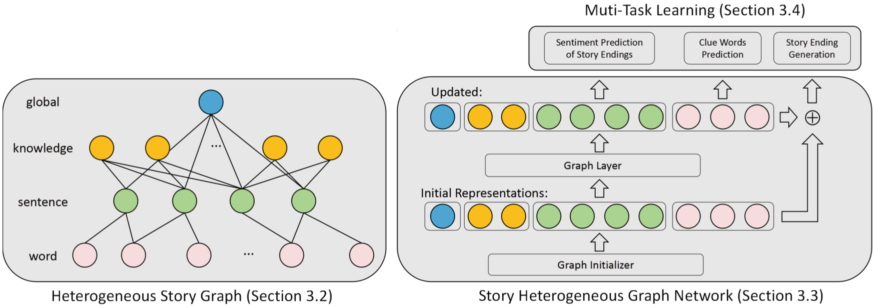

Muti-Task Learning (Section 3.4)
Sentiment Prediction of Story Endings
Clue Words Prediction
Story Ending Generation
global
knowledge
sentence
word
...
Updated:
Graph Layer
Initial Representations:
Graph Initializer
Heterogeneous Story Graph (Section 3.2)
Story Heterogeneous Graph Network (Section 3.3)

Fig. 2. Overview of our proposed model. For a story context, we first construct a heterogeneous story graph (Sect. 3.2). Then, we design our story heterogeneous graph network (SHGN) to initialize and update node representations in the graph (Sect. 3.3). Finally, auxiliary tasks are introduced to facilitate story comprehension, which are jointly optimized with the primary SEG task in the multi-task learning strategy (Sect. 3.4).

– Extensive experiments on widely used ROCStories Corpus show that our model achieves new state-of-the-art performances. Human study and case study further prove that our model could generate more coherent and reasonable story endings.

2 Related Work

Story Generation. Story Generation (SG), also known as storytelling, aims at generating a logical self-consistent story plot. Early SG work [5,23] mainly uses case-based or planning-based methods. Recently, researchers focus on generating stories with storyline or intermediate representations, such as skeletons [30], events [17], titles [12] and verbs [25]. In this way, they first generate intermediate representations, then rewrite and enrich them to obtain complete stories.

Story Ending Generation. Story Ending Generation (SEG) is a subtask of SG, which aims to understand the context and generate a coherent and reasonable story ending [33]. Li et al. [13] introduce Seq2Seq model with adversarial training to improve the rationality and diversity of the generated story endings. Similarly, Zhao et al. [33] employ Seq2Seq model based on reinforcement learning to generate more sensible endings. Gupta et al. [7] utilize an extra keywords extraction algorithm and model the keywords information in the proposed Seq2Seq model. Guan et al. [6] introduce a model which uses an incremental encoding scheme and commonsense knowledge to generate reasonable endings. Further, Huang et al. [10] propose a multi-level graph convolutional network to capture the dependency relations of input sentences. Although great progress has been made, the implicit knowledge and multi-grained interactive relations behind story context are still under-explored.

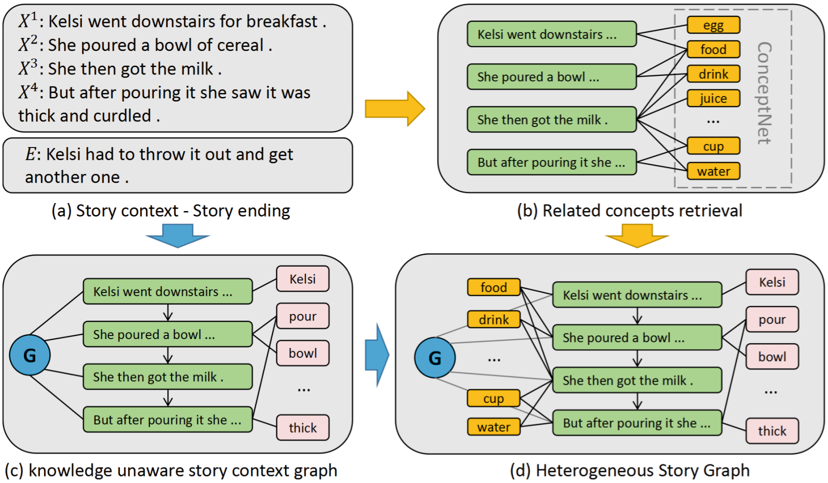

X^1: Kelsi went downstairs for breakfast .
X^2: She poured a bowl of cereal .
X^3: She then got the milk .
X^4: But after pouring it she saw it was thick and curdled .
E: Kelsi had to throw it out and get another one .
(a) Story context - Story ending
Kelsi went downstairs ...
She poured a bowl ...
She then got the milk .
But after pouring it she ...
egg
food
drink
juice
...
cup
water
ConceptNet
(b) Related concepts retrieval
G
Kelsi
pour
bowl
...
thick
(c) knowledge unaware story context graph
(d) Heterogeneous Story Graph

Fig. 3. Illustration of heterogeneous story graph construction process. Best viewed in color. Blue, orange, green and pink colors represent global, knowledge, sentence and word nodes, respectively.

3 Model

3.1 Overview

Story ending generation (SEG) task aims to generate a story ending conforming the corresponding context. Given a story context $X = \{X^1, X^2, ..., X^\mu\}$, where $X^k = x_1^k x_2^k ... x_l^k$ $(0 \leq k \leq \mu)$ represents the k-th sentence with l words. SEG aims at generating a story ending $E = y_1 y_2 ... y_m$ with m words.

Figure 2 shows the overview of our proposed model. To generate a coherent and reasonable story ending, we model the story context into a heterogeneous graph (Sect. 3.2). Based on the graph, we propose a Story Heterogeneous Graph Network (SHGN), which contains three components: (1) *graph initializer* is used to give each node an initial representation; (2) *graph layer* digests the structural information and gets updated node representations; (3) *transformer decoder* is used to generate the story endings according to final node representations (Sect. 3.3). Moreover, in order to sufficiently comprehend the story context, we design two auxiliary tasks, namely *sentiment prediction of story endings* and *clue words prediction*, which are expected to implicitly capture the sentiment trend as well as key events lie in context. These auxiliary tasks are jointly optimized with the primary story ending generation task in the multi-task learning strategy (Sect. 3.4).

3.2 Heterogeneous Story Graph Construction

We define the heterogeneous story graph as a directed graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R})$, where $v \in \mathcal{V}$ represents each node and $e \in \mathcal{E}$ denotes each edge. $\mathcal{A}$ and $\mathcal{R}$ are the sets of node types and edge types, respectively. $\tau(v) : \mathcal{V} \rightarrow \mathcal{A}$ and $\phi(e) : \mathcal{E} \rightarrow \mathcal{R}$ are type mapping functions which link nodes and edges to their specific type.

Figure 3 shows the construction process of heterogeneous story graph. For a given story context X, we utilize each word x_t^k of each context sentence X^k to retrieve related one-hop concepts from `ConceptNet` commonsense knowledge graph[2] [24] (Cf. Fig. 3(b)). To model the story context, we first construct knowledge unaware story graph by viewing sentences and words as different types of nodes. As shown in Fig. 3(c), each sentence node connects to both the next sentence (if has) and its word nodes (except stopwords). We also introduce a global node to aggregate non-local information. The global node connects to each sentence node using edges in both directions. Then, we combine the related concepts and knowledge unaware story graph as our heterogeneous story graph. Specifically, we only retain concepts retrieved from more than one context sentence to control the quality of retrieved concepts. The related concepts are regarded as knowledge nodes. If there are multiple identical knowledge nodes or word nodes, we also combine them into a single one. Figure 3(d) shows the overview of our final heterogeneous story graph which contains four types of nodes (global, knowledge, sentence, word) and seven types of edges (global ⇒ sentence, sentence ⇒ global, sentence ⇒ sentence, knowledge ⇒ sentence, sentence ⇒ knowledge, word ⇒ sentence, sentence ⇒ word) in total.

3.3 Story Heterogeneous Graph Network

The Story Heterogeneous Graph Network (SHGN) is used to initialize and update each node representation in the constructed graph (Sect. 3.2). Three components are introduced in SHGN: *graph initializer*, *graph layer* and *transformer decoder*.

Graph Initializer. The role of graph initializer is to give each node $v_i \in \mathcal{V}$ an initial representation $h_{v_i}^0$. For the global node, we randomly initialize its representation. For other nodes, the initial representations should contain the information of the corresponding documents, sentences, words or concepts. Owing to the emergence of large-scale pre-trained sentence embedding models [4,9,22], which show their superiority in many sentence-level NLP tasks, we decide to utilize SimCSE [4] (we also try different graph initializers in the experiments, and the SimCSE performs best. More details please refer to Sect. 4.4) to initialize representations. It is worth noting that the SimCSE model is also used to initialize knowledge and word nodes, instead of using geometric embedding methods (e.g., TransE) or word embedding methods (e.g., GloVe). In this way, there is no need to bridge the representation gap between different node types.

Graph Layer. Given a constructed graph $\mathcal{G}$ with node representations, we use Heterogeneous Graph Transformer (HGT) [8] as our graph layer, which models the heterogeneous graph by type-dependent parameters and can be easily applied to our heterogeneous story graph. Specifically, HGT includes:

[2] We only consider nouns, verbs, adjectives, and adverbs to retrieve related concepts from `ConceptNet`.

(a) *Heterogeneous Mutual Attention* is used to calculate the attention scores between source node s and target node t with the consideration of their edge $e = (s, t)$:

$$Attn^{(l)}(s, e, t) = \underset{\forall s \in N(t)}{softmax}(\alpha^{(l)}(s, e, t)) \tag{1}$$

$$\alpha^{(l)}(s, e, t) = (k_s^{(l)} W_{att,\phi(e)}^{(l)} q_t^{(l)^\top}) \tag{2}$$

$$k_s^{(l)} = W_{k,\tau(s)}^{(l)}(h_s^{(l-1)}) \tag{3}$$

$$q_t^{(l)} = W_{q,\tau(t)}^{(l)}(h_t^{(l-1)}) \tag{4}$$

where $N(t)$ denotes neighbors of target node t. $k_v^{(l)}/q_v^{(l)}$ represents the l-th layer key/query vector of node v. $h_v^{(l-1)}$ is the $(l-1)$-th layer representation of node v. $W_{att,\phi(e)}^{(l)}$, $W_{k,\tau(s)}^{(l)}$ and $W_{q,\tau(t)}^{(l)}$ are trainable parameters.

(b) *Heterogeneous Message Passing* is utilized to pass information from source nodes to target nodes:

$$Message(s, e, t) = \underset{i \in [1,h]}{||} MSGHead^i(s, e, t) \tag{5}$$

$$MSGHead^i(s, e, t) = W_{i,\tau(s)}^{(l)}(h_s^{(l-1)}) W_{\phi(e)}^{MSG} \tag{6}$$

where h is the number of heads in HGT, $W_{i,\tau(s)}^{(l)}$ and $W_{\phi(e)}^{MSG}$ are trainable parameters.

(c) *Target-Specific Aggregation* is used to aggregate information from the source nodes to the target node:

$$\tilde{h}_t^{(l)} = \underset{\forall s \in N(t)}{\oplus} (Attn^{(l)}(s, e, t) \cdot Message(s, e, t)) \tag{7}$$

$$h_t^{(l)} = W_{\tau(t)}(\sigma(\tilde{h}_t^{(l)})) + h_t^{(l-1)} \tag{8}$$

where $\oplus$ and σ represent addition operator and sigmoid function, respectively. $W_{\tau(t)}$ is trainable parameters.

After obtaining the output representation h_v^L for each node, we concatenate updated node representation h_v^L with corresponding initial representation h_v^0 and followed by a linear projection function to get the final node representation:

$$h_v^L = W_{final}[h_v^L, h_v^0] \tag{9}$$

Transformer Decoder. We utilize the vanilla Transformer decoder [26] to decode the story endings. The inputs of multi-head attention in transformer decoder are node representations $H^L = [h_{v_0}^L; h_{v_1}^L; ...; h_{v_s}^L]$ (s denotes the total number of nodes in the graph) and decoder input D_{in}. This process is denoted as:

$$\tilde{D}_{in} = MultiHead(D_{in}, H^L, H^L) \tag{10}$$

$$D_o = FFN(\tilde{D}_{in}) \tag{11}$$

where FFN is two linear projections with a ReLU activation in the middle. D_o is the middle output of transformer decoder.

To generate next word, a linear projection and softmax function are used to predict the word probabilities:

$$\mathcal{P}(y_t|y < t, X) = softmax(W_o D_o) \tag{12}$$

where W_o is trainable parameters, $\mathcal{P}(y_t)$ is the probability distribution over vocabulary.

Then, we calculate the negative data likelihood as loss function:

$$\mathcal{L}_{gen} = -\sum_t log\mathcal{P}(y_t = \tilde{y_t}|y < t, X) \tag{13}$$

3.4 Auxiliary Tasks

To sufficiently comprehend the story context and generate more coherent story endings, we design two auxiliary tasks with the special consideration for SEG: *sentiment prediction of story endings* and *clue words prediction.* These two auxiliary tasks are jointly optimized with the SEG task in a multi-task learning strategy.

Sentiment Prediction of Story Endings. Generally, the sentimental trend of the story context plays a crucial role in the SEG [18]. In order to improve the sentimental consistency of the generated endings. We make use of representations of all sentence nodes to predict the sentiment of the corresponding ending:

$$Y_s = W_s(\sum_{v_s} h^L_{v_s}) \tag{14}$$

where v_s denotes the sentence nodes and Y_s is the sentimental probability distribution.

We use the `VADER` toolkit [11] to construct sentiment labels of the gold story endings. The labels include "positive", "neutral" and "negative". During training, the cross entropy loss function can be defined as:

$$\mathcal{L}_{sen} = CE(Y_s, \tilde{Y_s}) \tag{15}$$

Clue Words Prediction. The story clues that lie in context are also important for SEG. Huang et al. [10] find that the words of top-2-degree in the dependency tree of each sentence are similar to the story clues summarized by humans. In detail, the dependency tree model the word sequences in the directed graph architecture, whose edges represent the dependency relations between words, such as causal relation, modifier relation, etc. The words of top-2-degree have the

Table 1. Statistics of ROCStories Corpus. Average, minimum, maximum and 95th percentile of length of datasets in wordpieces.

ROCStories corpus	Samples	Story context				Story ending			
		avg.	min	max	95ptcl.	avg.	min	max	95ptcl.
Training	90,000	35.0	4	65	48	9.5	1	20	14
Validation	4,081	32.6	15	56	46	8.9	2	18	13
Testing	4,081	33.7	12	61	47	9.4	2	17	14

most dependency relations with others. Thus, we utilize Biaffine [1] to construct the dependency tree for each sentence in the context and further regard the words of top-2-degree as clue words. We use the representation of each word node to predict whether it is a clue word. The binary cross entropy loss function of *clue words prediction* is denoted by:

$$\mathcal{L}_{clu} = CE(Y_w, \tilde{Y}_w) \tag{16}$$

$$Y_w = W_w h^L_{v_w} \tag{17}$$

where v_w denotes the word nodes and Y_w is the clue words probability distribution.

Multi-Task Learning. In our SHGN model, all three tasks all jointly performed through multi-task learning. The final objective is defined as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{sen} + \lambda_2 \mathcal{L}_{clu} + (1 - \lambda_1 - \lambda_2)\mathcal{L}_{gen} \tag{18}$$

where λ_1 and λ_2 are hyper-parameters.

4 Experiments

4.1 Experiment Setup

Dataset. Following previous work [6,10], we evaluate SHGN on the ROCStories Corpus [19] which contains 98,162 five-sentence daily life stories collected from crowd-workers. Table 1 shows the detailed statistics of the Corpus. In SEG task, the first four-sentence of each story is regarded as story context while the last sentence is the ground truth ending.

Implementation Details. We implement our model based on `transformers` [28] and `PyTorch Geometric` [3] libraries. For graph initializer, we utilize the SimCSE [4] sentence embedding model released by original authors[3], which has a similar architecture with `Robera-base` (768 hidden size, 12 multi-head attention, 12 layers). Following Feng et al. [29], the number of graph layer is set to 1

[3] https://huggingface.co/princeton-nlp/sup-simcse-roberta-base.

and we set the hidden size to 768. The transformer decoder used in our experiments has 12 decoder layers, 12 multi-head attention and 768 hidden size. To construct labels for sentiment prediction of story endings, we use `VADER` toolkit[4]. The dependency parsing algorithm Biaffine [1] used in our experiments is implemented by `SuPar`[5] toolkit.

During training, we set the batch size to 64 and use linear warmup of 1,000 steps. We employ grid search of Learning Rate (LR) in [2e−5, 3e−5, 5e−5] and number epochs in [5, 10, 15]. The best configuration used LR = 5e−5, 15 epochs. During inference, we use beam search and the beam size is set to 5. The coefficient λ_1 and λ_2 used in multi-task learning strategy are both 0.1.

Automatic Evaluation. We make use of BLEU [20] and ROUGE [14] for our automatic evaluation metrics, and report BLEU-1,2,3,4 together with ROUGE-1,2,L scores. Following previous study [10], we utilize `nlg-eval`[6] and `pyrouge`[7] toolkits to calculate the scores. Note that the BLEU or ROUGE scores might vary with different toolkits.

Human Evaluation. Considering the limitation of automatic evaluation, it is necessary to conduct human evaluation. Specifically, three aspects are considered as the criteria: (1) Grammaticality evaluates correct, fluent and natural of the generated story endings; (2) Logicality is used to evaluate whether the generated endings are reasonable and coherent; (3) Relevance measures how relevant are the endings to the story context. For a model, We randomly choose 100 generated story endings and employ three NLP postgraduates to make the evaluation. The scoring adopts a 3-point scale, with 1 as the worst and 3 as the maximum.

4.2 Baseline Methods

We compare our model with several typical baselines and the state-of-the-art baselines:

- **Seq2Seq+Att** [16]**:** A LSTM-based Seq2Seq model with attention mechanism.
- **Transformer** [26]**:** Transformers is a parallel Seq2Seq model based on multi-head attention and feed forward networks.
- **HLSTM** [31]**:** A hierarchical LSTM utilizes word-level and sentence-level LSTM as its encoder, and uses vanilla LSTM as its decoder to generate text sequence.
- **IE+MSA** [6]**:** A SEG model which considers external commonsense knowledge and uses an incremental encoding scheme to generate endings.
- **T-CVAE** [27]**:** A conditional variational auto-encoder model based on transformers.

[4] https://github.com/cjhutto/vaderSentiment.

[5] https://github.com/yzhangcs/parser.

[6] https://github.com/Maluuba/nlg-eval.

[7] https://github.com/bheinzerling/pyrouge.

Table 2. Experiments on the ROCStories Corpus for the SEG task. The **bold** denote the best performance. For performances of baseline methods, † represents the reproducing results while ‡ denotes the results reported by Huang et al. [10].

Model	B1	B2	B3	B4	R1	R2	R-L
Seq2Seq+Att [16]	$18.5^{\ddagger}$	$5.9^{\ddagger}$	–	–	–	–	–
Transformer [26]	$17.4^{\ddagger}$	$6.0^{\ddagger}$	–	–	–	–	–
HLSTM [31]	$22.1^{\ddagger}$	$7.1^{\ddagger}$	–	–	–	–	–
IE+MSA [6]	$24.3^{\dagger}$	$7.8^{\dagger}$	$3.9^{\dagger}$	$2.1^{\dagger}$	$17.5^{\dagger}$	$2.9^{\dagger}$	$20.8^{\dagger}$
T-CVAE [27]	$24.3^{\dagger}$	$7.7^{\dagger}$	$3.8^{\dagger}$	$2.0^{\dagger}$	$17.6^{\dagger}$	$3.0^{\dagger}$	$20.8^{\dagger}$
Plan&Write [32]	$24.4^{\dagger}$	$8.4^{\dagger}$	$4.1^{\dagger}$	$2.3^{\dagger}$	$18.1^{\dagger}$	$3.3^{\dagger}$	$21.4^{\dagger}$
MGCN-DP [10]	$24.5^{\dagger}$ ($24.6^{\ddagger}$)	$8.7^{\dagger}$ ($8.6^{\ddagger}$)	$4.3^{\dagger}$	$2.5^{\dagger}$	$18.4^{\dagger}$	$3.5^{\dagger}$	$21.9^{\dagger}$
SHGN (Our)	**25.6**	**9.4**	**4.7**	**2.7**	**20.3**	**3.9**	**23.5**

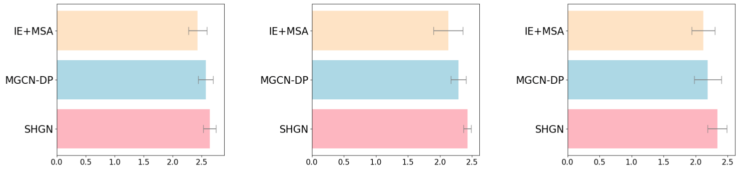
- **Plan&Write** [32]**:** A story generation model, which first uses a given title (topic) to obtain several keywords, and then generates complete stories.
- **MGCN-DP** [10]**:** The state-of-the-art SEG model which utilizes the dependency parse tree to construct a graph for story context, and makes use of GCN to capture story clues. A transformer decoder is employed to generate final endings.

Table 3. ROCStories Corpus ablations. "w/o" means "without". glob.: global, know.: knowledge, init.: initializer, SESP.: sentiment prediction of story endings, CWP.: clue words prediction. The **bold** and underline denote the best and the second performances, respectively.

#	Model	B2	B4	R-L
1	SHGN	**9.4**	**2.7**	**23.5**
2	SHGN (w/o glob.)	9.2	**2.7**	22.7
3	SHGN (w/o know.)	9.0	2.6	22.4
4	SHGN (w/o word)	8.9	2.6	22.3
5	SHGN (init. Sentence-BERT)	9.3	**2.7**	23.2
6	SHGN (init. LSTM)	7.6	2.3	21.2
7	SHGN (w/o mutli-task)	9.1	2.5	22.7
8	SHGN (w/o SPSE.)	9.3	2.6	23.1
9	SHGN (w/o CWP.)	9.2	2.5	22.9

4.3 Main Results

Table 2 shows the results of automatic evaluation. The results show that our model significantly outperforms these baselines. Specifically, our model



(a) Grammaticality (b) Logicality (c) Relevance

Fig. 4. Results on human evaluation, including means and variances. Our SHGN model outperforms IE+MSA and MGCN-DP on all three aspects.

achieves an improvement of 5.3%/5.3%/4.9%/4.5% over the IE+MSA/T-CVAE/Plan&Write/MGCN-DP in term of B1. As for B2, our model outperforms the IE+MSA/T-CVAE/Plan&Write/MGCN-DP by 20.5%/22.1% / 11.9%/8.0%, respectively. With respect to B4, our model implements an improvement of 28.6%/35.0%/17.4%/8.0%. And for R-L, our model achieves an improvement of 13.0%/13.0%/9.8%/7.3%. Other automatic metrics (i.e., B3, R1 and R2) also demonstrate the superiority of our SHGN. The results indicate that our model can comprehend the story context better based on the heterogeneous graph which model both the information of story context at different granularity (knowledge, sentence and word) levels and the multi-grained interactive relations among them.

4.4 Ablation Study

Effectiveness of Heterogeneous Graph. As described in Sect. 3.2, the constructed heterogeneous story graph contains four types of nodes: *global*, *knowledge*, *sentence* and *word*. We run 3 ablations, modifying various settings of our SHGN: (1) remove global node; (2) remove knowledge nodes; (3) remove word nodes. The effect of these ablations is shown in Table 3 (row 1 vs. row 2–4). In each case, the automatic evaluation scores are lower than our origin SHGN, which justifies the rationality of our model.

Effectiveness of Graph Initializer. We utilize SimCSE as our graph initalizer. We also run 2 ablations: (1) replace SimCSE [4] with Sentence-BERT [22]; (2) replace SimCSE with BiLSTM, where the forward and backward hidden states are concatenated as the initial node representations. GloVe.6B [21] word embedding (300 dimension) is used in the BiLSTM initializer. Table 3 (row 1 vs. row 5, 6) shows the effectiveness of SimCSE. Specifically, the results of BiLSTM initializer are dramatically dropped compared to pretrained sentence embedding models, which indicates the superiority of the pretrained models.

Effectiveness of Multi-Task Learning. In order to demonstrate the effectiveness of our auxiliary tasks, we remove each of them and all of them in ablation studies, respectively. The results are shown in Table 3 (row 1 vs. row 7–9), which indicates our designed auxiliary tasks can facilitate story context comprehension.

Table 4. Case Study on ROCStories Corpus. **Bold** words represent the key entities, events, or key words. *italic* words denote improper words.

	Case 1	Case 2
Context	Tim always had **stomach problems** He tried different things to **fix them** His doctors **couldn't** really **figure out** what was wrong It really **cut into** Tim's social **life**	Keith was **working at** a mechanic **shop** He had **given** a customer a **high quote** Keith **kept the difference** between the quote and the actual bill. Keith's boss **found out** what **he did**
IE+MSA	Tim was able to get a *new job*	Keith's boss was **mad** at him
MGCN-DP	Tim **felt better** after that	Keith *got* the **job**
SHGN	Tim eventually **got better**	Keith was **fired** from his **job**.
Gold	Eventually he learned to just **live with** the **discomfort**	Keith **lost** his **job**

4.5 Human Study and Case Study

We conduct human study on IE+MSA [6], MGCN-DP [10] and our SHGN. Figure 4 shows the results of human study. Our SHGN performs better than IE+MSA and MGCN-DP on all three aspects, which verifies that our SHGN performances better on generating reasonable and coherent story endings.

To provide a deeper understanding of generated endings, we show two examples from different models in Table 4. In Case 1, the IE+MSA model performs worst, since it generates a illogical story ending. In detail, the "*new job*" is irrelevant to the context. The MGCN-DP and SHGN output the endings with content about "*stomach problems*", which more relevant to the context, but they miss the context clue "*couldn't figure out*". This finding shows that: (1) The graph architecture could model the rich interactive relations across story and improve the relevance of generated endings; (2) SEG is still a challenging task. In Case 2, we know that there may be a bad ending for Keith since he did something illegal at work. However, MGCN-DP generates an unreasonable ending, which is contrary to the gold ending. IE+MSA and SHGN effectively capture the context clues and generate bad endings. Furthermore, the generated ending of SHGN is highly consistent with the gold ending, which shows the superiority of our model.

5 Conclusion

In this paper, we study SEG and propose Story Heterogeneous Graph Network (SHGN) which utilizes graph architecture and commonsense knowledge to comprehend the story context and handle the implicit knowledge behind the story. Besides, two auxiliary tasks are introduced to facilitate story comprehension. Extensive experiments on widely used ROCStories Corpus show that our SHGN achieves new state-of-the-art performances. Human study and case study further prove the effectiveness of our model.

Acknowledgement. This research is supported by the National Natural Science Foundation of China (Grant No. 62072323, 62102276), the Natural Science Foundation of Jiangsu Province (No. BK20191420, BK20210705, BK20211307), the Major Program of Natural Science Foundation of Educational Commission of Jiangsu Province, China (Grant No. 19KJA610002, 21KJD520005), the Priority Academic Program Development of Jiangsu Higher Education Institutions, and the Collaborative Innovation Center of Novel Software Technology and Industrialization. We would like to thank Duo Zheng and the members of the NLPTTG group, Netease Fuxi AI Lab for the helpful discussions and valuable feedback. We also thank anonymous reviewers for their suggestions and comments.

References

1. Dozat, T., Manning, C.D.: Deep biaffine attention for neural dependency parsing. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017, Conference Track Proceedings. OpenReview.net (2017). https://openreview.net/forum?id=Hk95PK9le
2. Feng, X., Feng, X., Qin, B., Geng, X.: Dialogue discourse-aware graph model and data augmentation for meeting summarization. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, pp. 3808–3814, August 2021. https://doi.org/10.24963/ijcai.2021/524. Main Track
3. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch geometric. ArXiv abs/1903.02428 (2019)
4. Gao, T., Yao, X., Chen, D.: SimCSE: simple contrastive learning of sentence embeddings. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 6894–6910. Association for Computational Linguistics, Punta Cana, November 2021. https://aclanthology.org/2021.emnlp-main.552
5. Gervás, P., Díaz-Agudo, B., Peinado, F., Hervás, R.: Story plot generation based on CBR. Know.-Based Syst. **18**(4–5), 235–242 (2005). https://doi.org/10.1016/j.knosys.2004.10.011
6. Guan, J., Wang, Y., Huang, M.: Story ending generation with incremental encoding and commonsense knowledge. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, 27 January–1 February 2019, pp. 6473–6480. AAAI Press (2019). https://doi.org/10.1609/aaai.v33i01.33016473
7. Gupta, P., Bannihatti Kumar, V., Bhutani, M., Black, A.W.: WriterForcing: generating more interesting story endings. In: Proceedings of the Second Workshop on Storytelling, pp. 117–126. Association for Computational Linguistics, Florence, August 2019. https://www.aclweb.org/anthology/W19-3413
8. Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. In: WWW 2020: The Web Conference 2020, Taipei, Taiwan, 20–24 April 2020, pp. 2704–2710. ACM/IW3C2 (2020). https://doi.org/10.1145/3366423.3380027
9. Huang, J., et al.: WhiteningBERT: an easy unsupervised sentence embedding approach. In: Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 238–244. Association for Computational Linguistics, Punta Cana, November 2021. https://aclanthology.org/2021.findings-emnlp.23
10. Huang, Q., et al.: Story ending generation with multi-level graph convolutional networks over dependency trees. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 14, pp. 13073–13081 (2021). https://ojs.aaai.org/index.php/AAAI/article/view/17545
11. Hutto, C., Gilbert, E.: Vader: a parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 8, no. 1, pp. 216–225, May 2014. https://ojs.aaai.org/index.php/ICWSM/article/view/14550

12. Li, J., Bing, L., Qiu, L., Chen, D., Zhao, D., Yan, R.: Learning to write stories with thematic consistency and wording novelty. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, 27 January–1 February 2019, pp. 1715–1722. AAAI Press (2019). https://doi.org/10.1609/aaai.v33i01.33011715
13. Li, Z., Ding, X., Liu, T.: Generating reasonable and diversified story ending using sequence to sequence model with adversarial training. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 1033–1043. Association for Computational Linguistics, Santa Fe, August 2018. https://www.aclweb.org/anthology/C18-1088
14. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81. Association for Computational Linguistics, Barcelona, July 2004. https://www.aclweb.org/anthology/W04-1013
15. Luo, F., et al.: Learning to control the fine-grained sentiment for story ending generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 6020–6026. Association for Computational Linguistics, Florence, Jul 2019. https://www.aclweb.org/anthology/P19-1603
16. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421. Association for Computational Linguistics, Lisbon, September 2015. https://www.aclweb.org/anthology/D15-1166
17. Martin, L.J., et al.: Event representations for automated story generation with deep neural nets. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, Louisiana, USA, 2–7 February 2018, pp. 868–875. AAAI Press (2018). https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17046
18. Mo, L., et al.: Incorporating sentimental trend into gated mechanism based transformer network for story ending generation. Neurocomputing **453**, 453–464 (2021)
19. Mostafazadeh, N., et al.: A corpus and cloze evaluation for deeper understanding of commonsense stories. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,pp. 839–849. Association for Computational Linguistics, San Diego, June 2016. https://www.aclweb.org/anthology/N16-1098
20. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics, Philadelphia, July 2002. https://www.aclweb.org/anthology/P02-1040
21. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. Association for Computational Linguistics, Doha, October 2014. https://www.aclweb.org/anthology/D14-1162
22. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992. Association for Computational Linguistics, Hong Kong, November 2019. https://www.aclweb.org/anthology/D19-1410
23. Riedl, M.O., Young, R.M.: Narrative planning: balancing plot and character. J. Artif. Int. Res. **39**(1), 217–268 (2010)

24. Speer, R., Havasi, C.: Representing general relational knowledge in ConceptNet 5. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), pp. 3679–3686. European Language Resources Association (ELRA), Istanbul, May 2012. http://www.lrec-conf.org/proceedings/lrec2012/pdf/1072_Paper.pdf
25. Tambwekar, P., Dhuliawala, M., Martin, L.J., Mehta, A., Harrison, B., Riedl, M.O.: Controllable neural story plot generation via reward shaping. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, 10–16 August 2019, pp. 5982–5988. ijcai.org (2019). https://doi.org/10.24963/ijcai.2019/829
26. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017). https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
27. Wang, T., Wan, X.: T-CVAE: transformer-based conditioned variational autoencoder for story completion. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, 10–16 August 2019, pp. 5233–5239. ijcai.org (2019). https://doi.org/10.24963/ijcai.2019/727
28. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45. Association for Computational Linguistics, October 2020. https://www.aclweb.org/anthology/2020.emnlp-demos.6
29. Xiachong, F., Xiaocheng, F., Bing, Q.: Incorporating commonsense knowledge into abstractive dialogue summarization via heterogeneous graph networks. In: Proceedings of the 20th Chinese National Conference on Computational Linguistics. pp. 964–975. Chinese Information Processing Society of China, Huhhot, August 2021. https://aclanthology.org/2021.ccl-1.86
30. Xu, J., Ren, X., Zhang, Y., Zeng, Q., Cai, X., Sun, X.: A skeleton-based model for promoting coherence among sentences in narrative story generation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4306–4315. Association for Computational Linguistics, Brussels, October-November 2018. https://www.aclweb.org/anthology/D18-1462
31. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489. Association for Computational Linguistics, San Diego, June 2016. https://www.aclweb.org/anthology/N16-1174
32. Yao, L., Peng, N., Weischedel, R., Knight, K., Zhao, D., Yan, R.: Plan-and-write: towards better automatic storytelling. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 7378–7385, July 2019. https://ojs.aaai.org/index.php/AAAI/article/view/4726
33. Zhao, Y., Liu, L., Liu, C., Yang, R., Yu, D.: From plots to endings: a reinforced pointer generator for story ending generation. In: Zhang, M., Ng, V., Zhao, D., Li, S., Zan, H. (eds.) NLPCC 2018. LNCS (LNAI), vol. 11108, pp. 51–63. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99495-6_5